DEBUG DEVICE

BACKGROUND OF THE INVENTION

TECHNICAL FIELD OF THE INVENTION

[0001] The present invention relates to a debug device using an on-chip debugging function, and more particularly to a debug device that can expand the break function using an external circuit.

CONVENTIONAL ART

[0002] Japanese laid-open patent application HEI 2-186448 (Japanese patent HEI 5-50016) describes an integrated circuit equipped with a debugging environment that enables software debugging by storing a primitive debugger program that enables debugging of the user program of a microcomputer ASIC while communicating with an external host computer in a ROM provided on the microcomputer ASIC chip.

[0003] Japanese laid-open patent application HEI 4-77833 (Japanese patent HEI 7-27472) describes an integrated circuit equipped with a debugging environment that enables debugging of programs stored in a user ROM and RAM externally, by performing the communication through a host computer via a serial communication block between a control CPU and the control ROM and RAM. .

[0004] Japanese laid-open patent application HEI 8-161191 describes an in-circuit emulator that can perform debugging even when a high-speed MPU is a target.

[0005] Japanese laid-open patent application HEI 8-179958 describes a microcomputer (controlling electronic device) that is capable of performing debugging

without using an ICE by executing a program for debugging assembled in the microcomputer according to a command from a host computer.

[0006] Japanese laid-open patent application HEI 10-214201 describes a microcomputer in which a debug circuit equipped with various functions used at the time of program debugging is provided in the microcomputer, and a flash memory electrically capable of writing/erasing incorporated on the same chip can be used as an emulation memory.

[0007] A magazine *Nikkei Electronics*, March 22, 1999 edition, at P215 – P225, describes the following in its article entitled "On-Chip Debugging Starts Penetrating In Assembly Software Development". Methods in developing software for embedded equipment internally equipped with microprocessors are changing. There are an increasing number of cases to employ an "on-chip debugging" method in which a microprocessor having a debugging exclusive command added in a command set and an exclusive debugger are combined for debugging software.

[0008] In the on-chip debugging, the break function is realized by using a break circuit or the like that is mounted on the chip. To realize the break function, a register to store break conditions, addresses for the CPU to access the break conditions set at the register, a comparator circuit to compare data and the like are needed. By providing a plurality of break circuit sets, a plurality of break points can be set. However, when a plurality of break circuit sets are provided, a circuit size (the number of gates) of the debug circuit section increases. There are instances that are not economically preferable to increase the circuit size more than required when realizing microcomputers or ASIC microcomputers. Accordingly, the break function (the number

of break points that can be set) that can be used in the on-chip debugging is generally limited to several (2 –8) points. In an actual debugging work, a map break function for detecting that the execution of a program extends to outside of an expected address range, data break functions under complex conditions and the like may be needed. In this respect, there has been a demand to expand a break function in a debugging device that uses an on-chip debugging function.

[0009] The present invention has been made to solve the problems described above. It is an object of the present invention to provide a debug device using an on-chip debugging function, in which the debug device uses an external circuit to expand a break function.

SUMMARY OF THE INVENTION

[00010] A debug device, in accordance with the present invention to solve the problems described above, is formed from a target board having a microcomputer equipped with an on-chip debugging function, a debugger that is connected via a debug interface terminal provided on the target board and that performs debugging using the on-chip debugging function implemented in the microcomputer, and a break board that monitors signals on an address bus, a data bus and a control bus which are led out from the target board, and outputs a break signal when a predetermined break condition is met.

[00011] It is noted that the break board may be equipped with a storage section to store break conditions, and break conditions may preferably be written in the storage section via the target board from the debugger.

[00012] In the debug device in accordance with the present invention, the break board is connected to the target board, and a break signal is output by the break board, with the result that the break function can be expanded. For example, even when an on-chip debugging function implemented in a microcomputer allows setting only several break points, many more break points can be set by using a break board. Accordingly, this realizes a variety of break functions, such as, for example, a map break in which the operation of a user program is interrupted by detecting that a specified address range is accessed, a bus break in which the operation of a user program is interrupted when specified data is written in or read from a specified address, a sequential break in which the operation of a user program is interrupted when a plurality of bus break conditions are generated in a specified sequence, and the like.

BRIEF DESCRIPTION OF THE DRAWINGS

[00013] Fig. 1 shows a block diagram of an overall structure of a debug device in accordance with the present invention.

[00014] Fig. 2 shows a block diagram of one example of a break board having a map break function.

[00015] Fig. 3 shows a block diagram of one example of a break board having a bus break function.

[00016] Fig. 4 shows a block diagram of one example of a break board having a sequential break function.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS OF THE PRESENT INVENTION

[00017] Embodiments of the present invention are described below with reference to the accompanying drawings.

[00018] Fig. 1 shows a block diagram of a general structure of a debugging device in accordance with the present invention. The debugging device 1 of the present invention is formed from a target board 10, a debugger 20 and a break board 30.

[00019] The target board 10 is provided with a microcomputer 11 equipped with an on-chip debugging function, a ROM 12, a RAM 13, and a variety of functional circuit sections (not shown). The target board 10 is provided with a group of debugging interface terminals 14 and a group of bus interface terminals 16 for externally leading out a variety of buses (address bus, data bus and control bus) 15.

[00020] The microcomputer 11 is equipped with at least a CPU 11a and a debug circuit 11b. The CPU 11a and the debug circuit 11b are connected by an internal bus (address bus, data bus, control bus) 11c. It is noted that the microcomputer 11 may be equipped with ROM, RAM, A/D converter, D/A converter, and peripheral functional circuits such as a variety of timer circuits, in addition to the CPU 11a and the debug circuit 11b.

[00021] An interface for debugging according to the Standard Test Access Port Boundary Scan Architecture (generally called as JTAG) is used. It is noted that the interface for debugging may have a structure that uses another interface.

[00022] The debug circuit 11b is equipped with a monitor ROM that stores a monitoring program for controlling debugging operations, a functional section that

performs data communication with the CPU 11a, a functional section that performs data communication with the debugger 20, a break circuit that temporarily stores break conditions for several points and generates a break signal when a specified break condition is detected to stop the execution of a user program, and a trace data extracting function section that extracts data required for real-time tracing in the execution state of the user program and transfers the extracted data to the side of the debugger 20.

[00023] A debug circuit 11b supplies a break request signal 11d to a non-maskable interrupt input terminal (a forced interrupt input terminal) of the CPU 11a to thereby stop the execution of the user program, when a break signal is generated by a break circuit provided in the debug circuit 11b, and when a break request is supplied from the side of the debugger 20.

[00024] The debugger 20 is formed form a debug tool 21 and a host system 22. The host system 22 is formed by using a personal computer, an engineering workstation or the like. Debugging software is installed in the host system 22. The debug tool 21 and the host system 22 are connected to one another by a host connection cable 23 such as an RS-232C cable or the like, such that data communication between the debug tool 21 and the host system 22 is executed through the host connection cable 23.

[00025] The debug tool 21 and the target board 10 are connected to one another by a cable for debugging 24. The debug tool 21 has a function to perform data communication with the debug circuit provided in the microcomputer 11 through the cable for debugging 24, and a function to perform data communication with the host system 22 through the host connection cable 23, and a function to control the debugging operations.

[00026] The debug tool 21 sends out a break command to stop the execution of the user program to the debug circuit 11b when a break generation signal 30a is supplied from the break board 30, such that a break request signal 11d is supplied through the debug circuit 11b to the non-maskable interrupt input terminal (forced interrupt input terminal) of the CPU 11a, to thereby stop the execution of the user program.

[00027] Specific operations of the debugger 20 and the debug circuit 11b are as follows. When a request for setting break points with respect to the break circuit provided in the debug circuit 11b is supplied from the host system 22, the debug tool 21 sends out a command to set the break points in the break circuit provided in the debug circuit 11b, whereby the break points are set in the break circuit provided in the debug circuit 11b.

[00028] When a request for executing a user program is supplied from the host system 22, the debug tool 21 supplies an execution command (e.g. "GO" command) of the user program to the debug circuit 11b such that the user program is executed.

[00029] Upon detection of a condition that is set as a break point during the execution of the user program, the debug circuit 11b supplies a break request signal 11d to the CPU 11a to thereby stop (break) the execution of the user program, and notifies to the debug tool 21 that the execution of the user program is broken. The debug tool 21 notifies to the host system 22 that the execution of the user program is broken. When a request to read content at a designated address is supplied from the host system 22, the debug tool 21 supplies the request to the debug circuit 11b to read the content at the designated address, and supplies the content read out to the host system 22.

[00030] The debug circuit 11b extracts debug data required for real-time tracing during the execution of the user program, and sends the extracted debug data to the debug tool 21. The debug tool 21 stores the debug data supplied from the debug circuit side in a trace memory section in the debug tool 21 in association with a time sequence, and supplies the debug data stored in the trace memory section to the host system 22.

[00031] Break board 30 is equipped with a break condition storage section 31, and a break signal generation section 32. The break condition storage section 31 is composed such that it can be accessed from the side of the CPU 11a through the various buses 15, and break conditions can be written and the written break conditions can be read from the side of the CPU 11a by designating previously set addresses. The break condition 31a stored in the break condition storage section 31 is supplied to the break signal generation section 32.

[00032] The break signal generation section 32 monitors the state of the variety of buses 15, and outputs a break generation signal 30a when the state of the variety of buses 15 coincides with a break condition 31a. The break generation signal 30a is supplied to the debug tool 21, and a break request signal 11d is supplied to the CPU 11a through the debug tool 21 and the debug circuit 11b.

[00033] It is noted that, when a serial communication is used for data communication between the debug tool 21 and the debug circuit 11b, there may be a time delay in supplying information about the generation of the break generation signal 30a to the debug circuit 11b. Therefore, when a serial communication is used for data communication between the debug tool 21 and the debug circuit 11b, the break

generation signal 30a may be directly supplied to the debug circuit 11b in a preferred structure. In this case, when a break generation signal 30a is supplied, the debug circuit 11b supplies a break request signal 11d to the CPU 11a to thereby stop the operation of the user program, and supplies information representing that a break is generated by the break board 30 to the debug tool 21.

[00034] Since the debug device 1 of the present invention has the structure described above, break conditions can be pre-set in the break condition storage section 31 in the break board 30, a break generation signal 30a can be generated when the break condition set by the break signal generation section 32 is met, and the execution of a user program can be broken based on the break generation signal 30a.

[00035] Fig. 2 shows a block diagram of one concrete example of a break board having a map break function. A break board 40 having a map break function shown in Fig. 2 is formed from a control section 41, a RAM 42 forming a break condition storage section, a data bus switching circuit 43, and a break signal generation section 44. The map break function is to stop (break) the execution of a user program, when a specified address, for example, an inappropriate address is accessed.

[00036] An address bus 15a is connected to a group of address input terminals 42a of the RAM 42 and also to a group of address input terminals 41a of the control section 41. A data bus 15b is connected to one group of terminals 43a of the data switching circuit 43 and also to a group of data input/output terminals 41b of the control section 41. Another group of terminals 43b of the data switching circuit 43 is connected to a group of data input terminals 44a of the break signal generation section 44. A group

of data input/output terminals 42b of the RAM 42 is connected to a group of common terminals 43c of the data switching circuit 43.

[00037] The control section 41 is equipped with an address decode circuit that outputs a signal indicating that, when an address previously set for the control section 41 is supplied, the address is designated, a data latch circuit that latches data supplied on the data bus 15b based on the output signal of the address decode circuit and a write signal, and a control circuit that controls the operation of the break board 40 based on the control data latched at the data latch circuit.

[00038] When a request to write control data is issued to the control section 41 from the CPU 11a side shown in Fig. 1 through the various buses 15 (when a request to write control data with previously set addresses being designated is issued to the control section 41), the control data is latched in an internal data latch circuit provided in the control section 41.

[00039] When the control data supplied from the CPU 11a side is a request to write a map break condition or a request to read a map break condition, the following operation takes place. The control section 41 outputs a data bus switching control signal 41c at, for example, H level. When the data bus switching control signal 41c at, for example, H level, is supplied, the data switching circuit 43 makes a condition in which the one group of terminals 43a and the group of common terminals 43c are connected to one another (a switched state indicated by a broken line in Fig. 2). As a result, the data bus 15b is connected to the data input/output terminals 42b of the RAM 42. The control section 41 supplies a write signal 15W on the control bus 15c to a write signal input

terminal 42c of the RAM 42, and supplies a read signal 15R on the control bus 15c to a read signal input terminal 42d of the RAM 42.

[00040] This makes a state in which the RAM 42 on the break board 40 becomes accessible through the various buses 15. In this state, data indicating whether or not an address is set as a break point is written at each of the addresses. For example, data "1" is written in an address where a break point is set, and data "0" is written in an address which is not set as a break point. It is noted that data stored in each of the addresses of the RAM 42 indicating whether or not it is a break point can be confirmed by reading out data stored in the RAM 42.

[00041] The debugger 20 writes data indicating whether or not it is a break point in the entire addresses of the RAM 42 on the break board 40 through the debug circuit 11b in the target board 10 and the CPU 11a, and when the above writing operation is completed, writes data requesting to start the map break operation in the control section 41.

[00042] The control section 41 outputs a data bus switching control signal 41c at, for example, L level, when the data requesting to start the map break operation is written. When the data bus switching control signal 41c at, for example, L level, is supplied, the data switching circuit 43 makes a condition in which the other group of terminals 43b and the group of common terminals 43c are connected to one another (a switched state indicated by a solid line in Fig. 2). As a result, the data input/output terminals 42b of the RAM 42 are connected to the data input terminals 44a of the break signal generation section 44.

[00043] Also, when the data requesting to start the map break operation is written, the control section 41 supplies a logical level signal indicating the read status to the read signal input terminal 42d of the RAM 42, and a logical level signal indicating that it is not a write state to the write signal input terminal 42c. Furthermore, when the data requesting to start the map break operation is written, the control section 41 supplies a break signal generation permission signal 41d that permits an operation of the break signal generation section 44 to the break signal generation section 44, such that the break signal generation section 44 is controlled to be placed in an active state (operation state)

[00044] As the user program is executed and various addresses are accessed, data indicating whether or not an address that is accessed is set as the break point is output from the RAM 42, and the data is supplied to the break signal generation section 44 through the data bus switching circuit 43.

[00045] The break signal generation section 44 determines whether or not it is a break point based on the data, and outputs a break generation signal 40a when it is the break point. In accordance with the present embodiment, when an address that is set as a break point is accessed, data "1" is output from the RAM 42. As a result, the break signal generation section 44 outputs a break generation signal 40a at, for example, H level, based on the data "1".

[00046] By using the break board 40 shown in Fig. 2 with the structure described above, a break point can be set for an address or each of multiple addresses, and break points can be set for one address range or a plurality of address ranges.

[00047] It is noted that the control section 41 may be structured to have a function to initialize data in the RAM 42, and data indicating a break point may be written only for an address that is designated as a break point.

[00048] In the present embodiment, an example in which data indicating whether or not an address is subject to a map break is stored in each of the addresses in the RAM 42 is shown. However, other structures may also be possible. For example, a plurality of data latch circuits may be provided to latch address that are set as break points, and a plurality of magnitude comparator circuits or the like are provided to detect a difference in the magnitude between an address (an address that is set as a break point) latched on the data latch circuit and an address on the address bus. A break signal may be generated at one address or each of a plurality of addresses based on comparison outputs from the plurality of magnitude comparator circuits or the like, or break generation signals may be generated in one address range or a plurality of address ranges.

[00049] Fig. 3 shows a block diagram of one concrete example of a break board equipped with a bus break function. The bus break function is a function to stop (break) the operation of the CPU when specified data is written in a specified address, or when specified data is read out from a specified address.

[00050] A break board 50 equipped with a bus break function shown in Fig. 3 is formed from a break condition setting control section 51, a break condition storage section 52, a break signal generation section 53, and a bus state latch section 54. The break condition storage section 52 is equipped with a break address storage section 52a, a break data storage section 52b, and a break generating condition storage section 52c. The

bus state latch section 54 is equipped with an address latch section 54a, a data latch section 54b, and an access state latch section 54c.

[00051] In the present embodiment, the bus width of the address bus 15a is 32 bits, and the bus width of the data bus 15b is 16 bits. The control bus 15c includes a write signal, a read signal, and an access space designation signal for discriminating between an access (write or read) for the memory address space and an access for an IO (input/output) address space.

[00052] Addresses that specify the break condition setting control section 51 are pre-set at the break condition setting control section 51. The break condition setting control section 51 is equipped with an address decode circuit which, when an address that is pre-set for the break condition setting control section 51 is supplied, outputs a signal indicating that the address is designated, and a control circuit that takes in data supplied on the data bus 15b based on an output signal of the address decode circuit and a write signal, and controls the setting of break conditions based on the data taken.

[00053] In accordance with the present embodiment, data for setting a bus break condition are supplied from the side of the debugger 20 shown in Fig. 1 through the target board 10 in the following order. First, control data for requesting to set a bus break condition is supplied; in the next bus cycle, data to designate an upper bit of the break address is supplied; in the next bus cycle, data to designate a lower bit of the break address is supplied; in the next bus cycle, break data is supplied; in the next bus cycle, data to designate a break generation condition is supplied; and in the next bus cycle, data to request the start of a bus break operation is supplied.

[00054] Here, a variety of break generation conditions as follows can be designated based on the data to designate a break generation condition. (1) A break takes place when designated data is written in a designated memory space address. (2) A break takes place when designated data is read from a designated memory space address. (3) A break takes place either in (1) or (2) above. (4) A break takes place when designated data is written in a designated IO space address. (5) A break takes place when designated data is read from a designated IO space address. (6) A break takes place either in (4) or (5) above.

[00055] When the control data for requesting to set a bus break condition is supplied, the break condition setting control section 51 recognizes a break address based on the data to designate an upper bit of the break address supplied in the next bus cycle and the data to designate a lower bit of the break address further supplied in the next bus cycle, and stores the break address in the break address storage section 52a. The break condition setting control section 51 stores break data supplied in the next cycle in the break data storage section 52b. The break condition setting control section 51 stores data to designate a break generation condition supplied in the next bus cycle in the break generating condition storage section 52c. When data to request the start of a bus break operation is supplied in the next cycle, the break condition setting control section 51 outputs a break signal generation permission signal 51a to control such that the break signal generation section 52 is placed in an operating state.

[00056] It is noted that, when data to request to stop the bus break operation is supplied, the break condition setting control section 51 stops the output of the break

signal generation permission signal 51a, and controls the break signal generation section 53 to be placed in a non-operating state.

[00057] The bus state latch section 54 latches a state of each of the buses at each bus cycle. More concretely, address data on the address bus 15a is latched at the address latch section 54a, data on the data bus 15b is latched at the data latch section 54b, and a state (writing in a memory space, reading from a memory space, writing in an IO space, and reading from an IO space) that is specified by signals in the control bus 15c is latched at the access condition latch section 54c.

[00058] In a state in which the break signal generation permission signal 51a is supplied, the break signal generation section 53 compares each break condition stored in the break condition storage section 52 and each latch data latched in the bus state latch section 54, and outputs a break generation signal 50a when each of the bus conditions coincides with each of the bus break conditions.

[00059] By the break board 50 equipped with a bus break function with the structure described above, a break generation signal 50a can be output when each of the buses 15a, 15b and 15c becomes a specified pre-set state.

[00060] It is noted that multiple sets of the circuits shown in Fig. 3 may be provided such that a plurality of bus break points can be set. In this case, a logical sum of break generation signals may be output as a break generation signal.

[00061] Fig. 4 shows a block diagram of one concrete example of a break board equipped with a sequential break function. The sequential break function is a function in which the execution of a user program is stopped when a plurality of bus break points are passed in a pre-set specified sequence. A break board 60 equipped with

a sequential break function shown in Fig. 4 is formed form n sets of bus break function circuit sections 50A – 50N, a bus break generation sequence designating data storage section 61 and a break signal generation section 62.

[00062] Each of the bus break function circuit sections 50A – 50N has a structure that is the same as that of the break board 50 equipped with a bus break function shown in Fig. 3.

[00063] Addresses that specify the bus break generation sequence designating data storage section 61 are pre-set at the bus break generation sequence designating data storage section 61. When a write request designating a specified address is supplied through the variety of buses, the bus break generation sequence designating data storage section 61 takes in bus break generation sequence designating data supplied on the data bus and stores the same. The bus break generation sequence designating data 61a stored in the bus break generation sequence designating data storage section 61 is supplied to the break signal generation section 62.

[00064] It is noted that bus break generation sequence data is written through the target board 10 from the side of the debugger 20 shown in Fig. 1. Also, each bus break condition is set in each of the bus break function circuit sections 50A – 50N through the target board 10 from the side of the debugger 20 shown in Fig. 1.

[00065] Here, a first bus break condition is set at the first bus break function circuit section 50A, a second bus break condition is set at the second bus break function circuit section 50B (not shown in the figure), and a third bus break condition is set at an n-th bus break function circuit section 50N. Also, the bus break generation sequence designating data storage section 61 stores conditions that cause to generate a break

generation signal 60a with respect to a sequential bus break when bus break points are passed in the order of the first bus break point, the second bus break point and the third bus break point.

[00066] The break signal generation section 62 monitors bus break generation signals 62a – 62n which are outputs of the respective bus break function circuit sections 50A – 50N, and outputs a break generation signal 60a when the generation sequence of the bus break generation signals 62a – 62n coincides with the generation sequence that is designated based on the bus break generation sequence designating data 61a.

[00067] Here, when the first bus break generation signal 62a indicating the detection of the first bus break point, the second bus break generation signal (not shown in the figure) indicating the detection of the second bus break point, and the third bus break generation signal 62n indicating the detection of the third bus break point are output in this sequence, and when the break signal generation section 62 outputs the third bus break generation signal 62n, the break generation signal 60a is output.

[00068] By using the break board 60 with the structure described above shown in Fig. 4, the operation of a user program can be interrupted when specified branch conditions are executed, or the operation of a user program can be interrupted when each of the various kind of input/output devices is accessed in a specified sequence. By this, the execution of a user program can be interrupted only under a complex branch condition or a specified access state, to execute debugging.

[00069] It is noted that a break board equipped with all of the functions of the break boards 40, 50 and 60 shown in Figs. 2 – 4 may be manufactured in order to cope

with each of the functions such as a map break function, a bus break function, and a sequential break function.

[00070] As described above, a debug device in accordance with the present invention is structured such that a break board is connected to a target board, and a break signal is generated by the break board. As a result, a debug device using an on-chip debugging function can expand its break function without being limited by the number of break circuits provided on the chip. Accordingly, debugging of user programs can be effectively conducted even with debug devices using the on-chip debugging function.

[00071] The entire disclosure of Japanese Patent Application No. 2000-403225 filed December 28, 2000 is incorporated by reference herein.